mathematics

Article

Machine Learning Control Based on Approximation of Optimal Trajectories

Askhat Diveev [1], Sergey Konstantinov [2], Elizaveta Shmalko [1,*] and Ge Dong [3]

[1] Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 119333 Moscow, Russia; aidiveev@mail.ru
[2] Department of Mechanics and Mechatronics, RUDN University, 117198 Moscow, Russia; svkonstantinov@mail.ru
[3] School of Aerospace Engineering, Tsinghua University, Beijing 100084, China; dongge@mail.tsinghua.edu.cn
* Correspondence: e.shmalko@frccsc.ru

Abstract: The paper is devoted to an emerging trend in control—a machine learning control. Despite the popularity of the idea of machine learning, there are various interpretations of this concept, and there is an urgent need for its strict mathematical formalization. An attempt to formalize the concept of machine learning is presented in this paper. The concepts of an unknown function, work area, training set are introduced, and a mathematical formulation of the machine learning problem is presented. Based on the presented formulation, the concept of machine learning control is considered. One of the problems of machine learning control is the general synthesis of control. It implies finding a control function that depends on the state of the object, which ensures the achievement of the control goal with the optimal value of the quality criterion from any initial state of some admissible region. Supervised and unsupervised approaches to solving a problem based on symbolic regression methods are considered. As a computational example, a problem of general synthesis of optimal control for a spacecraft landing on the surface of the Moon is considered as supervised machine learning control with a training set.

Keywords: machine learning control; general synthesis problem; symbolic regression; optimal control; evolutionary algorithm

check for updates

Citation: Diveev, A.; Konstantinov, S.; Shmalko, E.; Dong, G. Machine Learning Control Based on Approximation of Optimal Trajectories. *Mathematics* **2021**, *9*, 265. https://doi.org/10.3390/math9030265

Academic Editor: Mikhail Posypkin

Received: 30 December 2020
Accepted: 26 January 2021
Published: 29 January 2021

1. Introduction

Complexity of the control synthesis problems for autonomous robots which must perform the assigned tasks and achieve the set goal, led to new ideas in the control theory. Now, to create a control system for an autonomous robot, this system needs to be trained [1,2], instead of obtaining it by solving some known optimization problems.

To formulate the real problem of mobile robot control, it is needed to describe a large number of different phase constraints. These can be walls, doors between the rooms, windows, columns and other obstacles. For example, a robot has to avoid a column, not to hit on a wall and to get in a door. Now, when control systems for mobile robots are being created, programmers imagine the problems that this robot must face and decide how it should overcome them. Quite a laborious process, but it is quite justified in conditions when control systems were developed on an individual basis for single technical objects, such as spacecraft. However, modern automation and robotization is reaching a broader level and becoming ubiquitous. This trend requires the development of new universal and even automatic approaches to the development of control systems.

Application of symbolic regression methods allows to automatically receive mathematical expressions for control functions. Such mathematical expressions describe how the robot should optimally reach the goal avoiding the obstacles.

Only symbolic regression methods can search structure and parameters of mathematical expression. Other methods, and even artificial neural networks, search only parameters.

The searching of control function structure in the control synthesis problem is called machine learning control [1]. This is a new technology in the development of control systems and it has not yet been proposed a rigorous mathematical formulation that defines this approach. In this paper, we propose some mathematical formalization of the machine learning problem (Section 2) and, on the basis of the proposed definitions, we single out a special area of machine learning—machine learning control (Section 3).

One of the main problems of machine learning control is the problem of control synthesis. The paper first presents the general mathematical formulation of the control synthesis problem, and then proposes its numerical formulation, since according to the methodology of machine learning control, the synthesis problem must be solved numerically using symbolic regression methods.

Further in the work in Section 4, we present our approach to solving the problem of machine learning control based on approximation of optimal trajectories. According to the technique of learning firstly it is necessary to create a training set in order to show to learning object what we want of it. For this purpose initially the optimal control problem is solved with the same quality criterion as for the synthesis problem from some different initial conditions. Obtained optimal trajectories are templates for learning. They show what forms of plots for variables must be obtained in the result of control synthesis problem solution and what values of functional must give these solutions. Then, obtained optimal trajectories for different initial conditions are approximated by a numerical method of symbolic regression. The proposed approach of machine learning based on approximation of optimal trajectories is demonstrated in the computational example of general synthesis of optimal control for a spacecraft landing on the surface of the Moon (Section 5).

2. Problem Statement of Machine Learning

Definition 1. *A set of computational procedures, that transforms a vector* $\mathbf{x}$ *from an input space* X *to a vector* $\mathbf{y}$ *from an output space* Y*, and there is not any algebraic equation* $\mathbf{y} = \mathbf{f}(\mathbf{x})$ *for them, is called an* **unknown function.**

For example, the system of ordinary differential equations $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ is an unknown function for a vector of initial conditions $\mathbf{x}(0)$ and a vector of solutions as functions of time $\mathbf{x}(t, \mathbf{x}(0))$, if a general solution is unknown for this differential equation.

The unknown function between input vector $\mathbf{x}$ and output vector $\mathbf{y}$ is defined as

$$\mathbf{y} = \alpha(\mathbf{x}). \tag{1}$$

Then for differential equations without general solutions, an unknown function has a form

$$\mathbf{x}(t, \mathbf{x}^0) = \alpha(\mathbf{x}^0). \tag{2}$$

Definition 2. *A work area is a subset of input vector space, where the input vectors exist surely and that is used for solving the problem.*

The unknown function can be realized by a physical equipment or an experiment. Then unknown function will be called black box, but will be described as (1).

Let a set of input vectors be determined in the work area

$$\tilde{\mathrm{X}} = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\} \subseteq \mathrm{X}. \tag{3}$$

For every input vector, output vector is determined by the unknown function (1)

$$\tilde{\mathrm{Y}} = \{\mathbf{y}^1 = \alpha(\mathbf{x}^1), \ldots, \mathbf{y}^N = \alpha(\mathbf{x}^N)\} \subseteq \mathrm{Y}. \tag{4}$$

Definition 3. *A pair of sets,*

$$(\tilde{\mathrm{X}}, \tilde{\mathrm{Y}}), \tag{5}$$

is called a training set.

It is known that there are supervised and unsupervised machine learning methods.

An **unsupervised machine learning problem** can be formulated as follows: for some unknown function (1) and a positive small value δ it is necessary to find a function

$$\mathbf{y} = \beta(\mathbf{x}, \mathbf{q}), \tag{6}$$

where $\mathbf{q}$ is a vector of parameters, $\mathbf{q} = [q_1 \dots q_p]^T$, such that $\forall \mathbf{x} \in \mathrm{X}$

$$\|\mathbf{y} - \beta(\mathbf{x}, \mathbf{q})\| \leq \delta. \tag{7}$$

A **supervised machine learning problem** consequently can be formulated as follows: for some unknown function (1) and a positive small value δ, it is necessary to determine a positive value ε, to build a training set (5) and to find a function (6) such that if the total error for the training sample is less than the given value ε

$$\sum_{i=1}^{N} \|\mathbf{y}^i - \beta(\mathbf{x}^i, \mathbf{q})\| \leq \varepsilon, \tag{8}$$

then for $\forall \mathbf{x}^*$ from work area, but not included in the training set $\mathbf{x}^* \in \mathrm{X}$ and $\mathbf{x}^* \notin \tilde{\mathrm{X}}$ the following inequation is performed

$$\|\mathbf{y}^* - \beta(\mathbf{x}^*, \mathbf{q})\| \leq \delta, \tag{9}$$

where $\mathbf{y}^* = \alpha(\mathbf{x}^*)$.

Here the function $\beta(\mathbf{x}, \mathbf{q})$ includes a parameter vector $\mathbf{q}$. In many approaches a structure of function is defined beforehand on the basis of experience or intuitively, and it is necessary to find only values of some parameters. For example, an artificial neural network [3–5], which is often used for solving of the machine learning problems, has a set structure and large number of unknown parameters. In contrast, symbolic regression methods [6–8] allow you to search for both function structure and parameters.

3. The Problem of General Control Synthesis as Machine Learning Control

In the field of control there are also problems that require machine learning. One of the main machine learning control problems is a search for a control function in the general control synthesis problem.

The problem of control general synthesis was formulated in the middle of the last century by Boltyanskii [9] after studying the Pontryagin's maximum principle for the optimal control problem.

The problem has the following description.

The mathematical model of the control object is given in the form of the system of ordinary differential equations

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \tag{10}$$

where $\mathbf{x}$ is a vector of state, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u}$ is a vector of control, $\mathbf{u} \in \mathrm{U} \subseteq \mathbb{R}^m$, U is a compact set, $m \leq n$.

The domain of initial conditions is given

$$\mathrm{X}_0 \subseteq \mathbb{R}^n. \tag{11}$$

Existence of the initial condition domain is a main feature of the control general synthesis problem. Initially Boltyanskii defined the domain of initial conditions as a whole space of states $\mathrm{X}_0 = \mathbb{R}^n$, because he tried to solve this problem analytically. In this case we assume to solve this problem numerically. Therefore the domain X_0 is a restricted set in the space of states.

The terminal condition is given

$$\mathbf{x}(t_f) = \mathbf{x}^f \in \mathbb{R}^n, \tag{12}$$

where t_f is unassigned time of getting from any initial condition $\mathbf{x}^0 \in \mathrm{X}_0$ to the terminal state (12).

The finishing time is bounded

$$t_f \leq t^+, \tag{13}$$

where t^+ is a given positive value.

The phase constraints are given

$$\varphi_i(\mathbf{x}) \leq 0, \ i = 1, \ldots, r. \tag{14}$$

The quality criterion is given

$$J = \int\limits_{\mathrm{X}_0} \cdots \int \int\limits_{t_0}^{t_f} f_0(\mathbf{x}(t, \mathbf{x}^0), \mathbf{u}(t))dt \to \min_{\mathbf{u} \in \mathrm{U}}, \tag{15}$$

where $\mathbf{x}(t, \mathbf{x}^0)$ is a partial solution of differential Equation (10) with control $\mathbf{u}(t) \in \mathrm{U}$ from initial condition $\mathbf{x}^0 \in \mathrm{X}_0$.

It is necessary to find a control function in the form

$$\mathbf{u} = \mathbf{h}(\mathbf{x}) \in \mathrm{U}, \tag{16}$$

where $\mathbf{h}(\mathbf{x}) : \ \mathbb{R}^n \to \mathbb{R}^m$.

If one inserts the control function (16) in the right part of differential Equation (10), then the system of stationary differential equations is received

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{h}(\mathbf{x})), \tag{17}$$

which does not have a free control vector in the right part.

Any partial solution of the differential Equation (17) from initial conditions (11) achieves terminal condition (12), performing all conditions on phase constraints (14) with optimal value of the quality criterion (15).

Note, that the control function (16) can have simple discontinuities, therefore in many cases analytical methods could not be applied. The majority of analytical methods such as integrator backstepping [10,11] and analytical design of aggregated regulators [12,13] provides stability on Lyapunov by nonlinear smooth feedback control. The main drawback of all analytical methods of control synthesis solution is that they are bounded with the specific form of the mathematical model of control object. The control synthesis problem (10)–(17) under consideration is complicated by the arbitrary form of the mathematical model of the control object and sub-integral function of quality criterion, as well as the phase constraints and a wide class of control functions, which can have simple discontinuities.

In general case, this control general synthesis problem can be solved numerically by symbolic regression methods as machine learning control problem.

For application of the numerical methods it is necessary to reformulate the problem statement. The domain of initial conditions is changed onto finite set of initial state points

$$\tilde{\mathrm{X}}_0 = \{\mathbf{x}^{0,1}, \ldots, \mathbf{x}^{0,K}\}. \tag{18}$$

The terminal condition (12) and the phase constraints are added into quality criterion (15), and the integral of the domain of initial conditions is changed onto sum of all initial state points.

$$J_1 = \sum_{i=1}^{K} \Big(a_1 \|\mathbf{x}^f - \mathbf{x}(t_{f,i}, \mathbf{x}^{0,i})\| + \int\limits_0^{t_{f,i}} \Big(f_0(\mathbf{x}(t, \mathbf{x}^{0,i}), \mathbf{u}(t)) +$$

$$\vartheta(\varphi(\mathbf{x}(t, \mathbf{x}^{0,i})))p(\mathbf{x}(t, \mathbf{x}^{0,i}))\Big) dt \Big) \to \min_{\mathbf{u} \in \mathrm{U}}, \tag{19}$$

where a_1 is a weight coefficient, $\vartheta(A)$ is a Heaviside step function

$$\vartheta(A) = \begin{cases} 1, \text{if } A > 0 \\ 0, \text{otherwise} \end{cases}, \tag{20}$$

$p(B)$ is a penalty function, $t_{f,i}$ is a time of terminal state (12) achievement from initial condition $\mathbf{x}^{0,i}$,

$$t_{f,i} = \begin{cases} t, \text{if } t \le t^+ \text{and } \|\mathbf{x}^f - \mathbf{x}(t, \mathbf{x}^{0,i})\| \le \varepsilon_0 \\ t^+, \text{otherwise} \end{cases}, \ i = 1, \ldots, K, \tag{21}$$

ε_0 is a small positive value, that determines accuracy of terminal state achievement.

Within the framework of the formulation of the machine learning problem, the solution to the synthesis problem based on symbolic regression methods is machine learning control.

3.1. Control Synthesis as Unsupervised Machine Learning Control

The first approach is a direct search of the control function on basis of a quality criterion minimization. In this case we receive unsupervised machine learning control. The stated general control synthesis problem (10), (12), (18)–(21) can be solved in the concept of unsupervised machine learning control by different symbolic regression methods. Such approach is demonstrated by genetic programming [2], network operator method [14], variational genetic programming [15], variational analytic programming [16], multi-layer network operator [17], binary variational genetic programming [18], modified Cartesian genetic programming [19]. All mentioned symbolic regression methods search for mathematical expressions of control functions, that provide for the received solutions achievement of the terminal condition (12) from all the initial conditions (18) with optimal value of the quality criterion (19), describing the time and accuracy of terminal state hitting, and including phase constraints in the form of penalty functions.

Symbolic regression methods use evolutionary algorithms to search for functions and can achieve a certain level of accuracy when minimizing the functional, but it still remains unknown how the values of the criterion (19) for these solutions are far from real optimal values. To correct this problem it is possible to use a supervised machine learning with a training set received by the solution of the optimal control problem.

3.2. Control Synthesis as Supervised Machine Learning Control

The second approach is a learning with application of a training set. This is a supervised machine learning control. In this case firstly it is necessary to obtain the training set. For this purpose solutions of the optimal control problem can be used.

The statement of optimal control problem includes a mathematical model of control object (10), an initial condition given in one point

$$\mathbf{x}^0 \in \mathbb{R}^n, \tag{22}$$

terminal condition (12), (13) the phase constraints (14), and a quality criterion

$$J_2 = a_1\|\mathbf{x}^f - \mathbf{x}(t_f)\| + \int_0^{t_f} (f_0(\mathbf{x}(t), \mathbf{u}(t)) + \vartheta(\varphi(\mathbf{x}(t))p(\mathbf{x}(t)))dt \to \min_{\mathbf{u}\in\mathrm{U}}, \tag{23}$$

where

$$t_f = \begin{cases} t, \text{if } t \leq t^+ \text{and } \|\mathbf{x}^f - \mathbf{x}(t)\| \leq \varepsilon_0 \\ t^+, \text{otherwise} \end{cases}. \tag{24}$$

It is necessary to find a control in the form

$$\tilde{\mathbf{u}} = \mathbf{v}(t, \mathbf{x}^0) \in \mathrm{U}. \tag{25}$$

When inserting the function (25) into the right part of the mathematical model of the control object (10), the following system of non-stationary differential equations is received

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{v}(t, \mathbf{x}^{0,i})). \tag{26}$$

To create a training set for the control synthesis problem it is necessary to solve the optimal control problem on criterion (23) for each particular initial condition from (18) and to receive sets of optimal controls

$$\mathrm{U}_0 = \{\mathbf{v}(t, \mathbf{x}^{0,1}), \ldots, \mathbf{v}(t, \mathbf{x}^{0,K})\} \tag{27}$$

and optimal trajectories

$$\tilde{\mathrm{X}} = \{\tilde{\mathbf{x}}(t, \mathbf{x}^{0,1}), \ldots, \tilde{\mathbf{x}}(t, \mathbf{x}^{0,K})\}. \tag{28}$$

Now we define the time interval $Deltat$ and calculate the value of the state vector on each optimal trajectory at the interval boundaries. As a result, get a training set of optimal trajectories

$$\tilde{\mathrm{X}} = \{\tilde{\mathrm{X}}_1, \ldots, \tilde{\mathrm{X}}_K\}, \tag{29}$$

where

$$\tilde{\mathrm{X}}_i = \{\tilde{\mathbf{x}}(0, \mathbf{x}^{0,i}) = \mathbf{x}^{0,i}, \tilde{\mathbf{x}}^i(t_1, \mathbf{x}^{0,i}), \ldots, \tilde{\mathbf{x}}^i(t_{M_i}, \mathbf{x}^{0,i})\},\ i = 1, \ldots, K, \tag{30}$$

$t_j = t_{j-1} + \Delta t$, $j = 1, \ldots, M_i$, $i = 1, \ldots, K$, Δt is a given time interval.

Now in order to solve the control synthesis problem (10), (12), (18)–(21), and to find the control function in the form (16) it is enough to approximate the training set (29) on a criterion

$$J_3 = \sum_{i=1}^{K} \sum_{j=0}^{M_i} \|\mathbf{x}(t_j, \mathbf{x}^{0,i}) - \tilde{\mathbf{x}}(t_j, \mathbf{x}^{0,i})\| \to \min_{\mathbf{h}(\mathbf{x})\in\mathrm{U}}, \tag{31}$$

where $t_0 = 0$, $\mathbf{x}(t, \mathbf{x}^{0,i})$ is a partial solution of the Equation (17) with the initial conditions $\mathbf{x}^{0,i}$, $\tilde{\mathbf{x}}(t, \mathbf{x}^{0,i})$ is a partial solution of the Equation (28), $i \in \{1, \ldots, K\}$.

To ensure the fulfillment of phase constraints, both criteria (31) and (19) are applied. In result, the following combined criterion is used

$$J_4 = J_1 + \gamma J_3 \to \min_{\mathbf{h}(\mathbf{x})\in\mathrm{U}}, \tag{32}$$

where γ is a weight coefficient.

To solve the approximation problem, a symbolic regression is also used. The control synthesis on the base of optimal trajectories approximation allows to find a control function (16) that provides receiving optimal control with accuracy to approximation of the training set. The solution closest to the optimal one is determined by the accuracy of the optimal control problem.

4. Computational Algorithms

In order to solve the control synthesis problem as machine learning control on the base of optimal trajectories set approximation it is required to solve two complex problems, the optimal control problem in order to form a training set, and the approximation of optimal trajectories by some symbolic regression method. For both problems evolutionary computations are used.

4.1. Algorithms for the Optimal Control Problem

The optimal control problem with phase constraints is not uni-modal [20], therefore evolutionary algorithms are applied, which can solve a global optimization problem. Recently, it is popular to use hybrid evolutionary algorithms that combine different evolutionary algorithms. Studies of evolutionary algorithms for numerical solution of the optimal control problem show [21], that the most successful in solving this problem are genetic algorithm (GA) [22], Particle swarm optimization algorithm (PSO) [23] and Grey wolf optimizer algorithm (GWO) [24].

All evolutionary algorithms include evolution of possible solutions. It implies such changes in possible solutions that some of new possible solutions ensure obtaining the value of the goal functional not worse than the old possible solutions before the change. At evolution of possible solutions information about the values of the goal functional for other possible solutions is used.

PSO-algorithm uses the best current possible solution, and the best of solutions from some random selected ones as well as information about historical changes for this possible solution. An evolution is performed for each component of possible solution

$$\tilde{q}_j^i = q_j^i + \sigma v_j^i, \ \ j = 1, \dots, p, \tag{33}$$

where $\tilde{q}_j^i$ is a new value of the component j of the possible solution i, $\tilde{\mathbf{q}}^i = [\tilde{q}_1 \dots \tilde{q}_p]^T$, q_j^i is the component j of the old possible solution i, $\mathbf{q}^i = [q_1^i \dots q_p]^T$, p is a dimension of the vector, v_j^i is the component j of a historical vector for possible solution i, $\mathbf{v}^i = [v_1^i \dots v_p^i]$, this historical vector is changed one time in one cycle of generation

$$v_j^i \leftarrow \alpha v_j^i + \xi\beta(q_j^i(k) - q_j^i) + \xi\gamma(q_j^0 - q_j^i), j = 1, \dots, p, \tag{34}$$

$\bar{q}_j^i$ is the component j of the best solution from k randomly selected ones, $\bar{\mathbf{q}}^i = [\bar{q}_1 \dots \bar{q}_p]^T$, $q_j(0)$ is the component j of the best current possible solution, $\mathbf{q}(0) = [q_1(0) \dots q_p(0)]^T$, ξ is a random value in the interval from 0 to 1, at each call this function gives a new random number, α, β, γ are constant parameters of the algorithm, vector $\mathbf{v}$ has zero initial value.

The GWO-algorithm performs the following changes of possible solution on the base of some best current solutions

$$\tilde{q}_j^i = \frac{1}{N}\sum_{k=0}^{N-1}(q_j(k) - r(2\xi - 1)|2\xi q_j(k) - q_j^i|), \ \ j = 1, \dots, p, \tag{35}$$

where $q_j(0) \dots q_j(N-1)$ are j components of N best possible solutions,

$$r = 2\left(1 - \frac{g}{G}\right), \tag{36}$$

r is calculated one time in a generation, g is a number of generation, G is a quantity of generations.

The GA considers vectors in the form of Grey code and performs evolution for two selected possible solutions as operations of crossover and mutation. For crossover two possible solutions are selected

$$\mathbf{z}^i = [z_1^i \dots z_{c+d}^i]^T, \ \ \mathbf{z}^j = [z_1^j \dots z_{c+d}^j]^T, \tag{37}$$

where $z_k^i, z_k^j \in \{0,1\}$, c is a number of bit for integer part, d is a number of bits for fractional part of Grey code.

Then the point of crossover s is determined. As a result two new possible solutions are received

$$\tilde{\mathbf{z}}^i = [z_1^i \dots z_s^i \; z_{s+1}^j \dots z_{c+d}^j]^T, \;\; \tilde{\mathbf{z}}^j = [z_1^j \dots z_s^j \; z_{s+1}^i \dots z_{c+d}^i]^T. \tag{38}$$

In hybrid algorithms in each cycle of evolution for each possible solution one of three ways (33), (34) or (35), (36), or (37), (38) of obtaining new possible solutions is selected randomly. The algorithm stops calculation after all cycles of evolution are performed.

4.2. Numerical Methods of Symbolic Regression for the Control Synthesis Problem

For solution of the control synthesis problem numerical methods of symbolic regression are used. Now more than fourteen methods are know. The methods code a mathematical expression and search for optimal solution on the code space. All methods differ in the form of code.

For example, consider the following mathematical expression

$$y = \sin(x_1) + \exp(-q_1 x_1)\cos(q_2 x_2 + q_1). \tag{39}$$

To code this mathematical expression the following basic sets are used:

the set of arguments

$$\mathrm{F}_0 = \{f_{0,1} = x_1, f_{0,2} = x_2, f_{0,3} = q_1, f_{0,4} = q_2\}, \tag{40}$$

the set of elementary functions

$$\mathrm{F} = \{f_{1,1}(z) = z, f_{1,2}(z) = -z, f_{1,3}(z) = \sin(z), f_{1,4}(z) = \cos(z),$$

$$f_{1,5}(z) = \exp(z), f_{2,6}(z_1, z_2) = z_1 + z_2, f_{2,7}(z_1, z_2) = z_1 z_2\}, \tag{41}$$

where indexes of elements point the number of arguments and a function number, if the first index is equal to zero, then this is an argument of the mathematical expression.

The most popular and the earliest symbolic regression method is the genetic programming by J.Koza [6]. This method presents a mathematical expression in the form of computational tree. In the Figure 1 the computational tree for the mathematical expression (39) is presented.

In the Figure 1 the nodes of the tree contain numbers of functions, the leaves contain arguments of the mathematical expression.

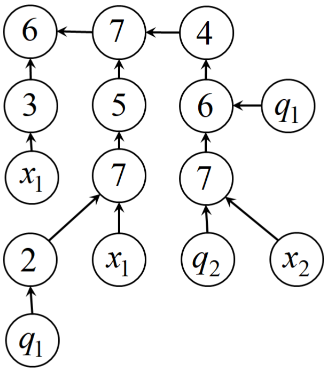

Figure 1. Computational tree of genetic programming.

A code of genetic programming for the mathematical expression (39) has the following form

$$\left(\begin{bmatrix}2\\6\end{bmatrix},\begin{bmatrix}1\\3\end{bmatrix},\begin{bmatrix}0\\1\end{bmatrix},\begin{bmatrix}2\\7\end{bmatrix},\begin{bmatrix}1\\5\end{bmatrix},\begin{bmatrix}2\\7\end{bmatrix},\begin{bmatrix}1\\2\end{bmatrix},\begin{bmatrix}0\\3\end{bmatrix},\begin{bmatrix}0\\1\end{bmatrix},\right.$$
$$\left.\begin{bmatrix}1\\4\end{bmatrix},\begin{bmatrix}2\\6\end{bmatrix},\begin{bmatrix}2\\7\end{bmatrix},\begin{bmatrix}0\\2\end{bmatrix},\begin{bmatrix}0\\4\end{bmatrix},\begin{bmatrix}0\\3\end{bmatrix}\right). \tag{42}$$

The code of genetic programming consists of indexes of elements of the computational tree on all branches from the top to the leaves. The code of genetic programming is used for presentation of the computational tree in the computer memory.

A code of genetic programming is not very comfortable, as codes of different mathematical expressions have different length that also changes after crossover operation. If in the mathematical expression one argument enters several times, then it has to be on leaves of the computational tree the same number of times.

Another method of symbolic regression—the network operator method [14]—codes mathematical expression in the form of oriented graph. In the Figure 2 the network operator graph for the mathematical expression (39) is presented.

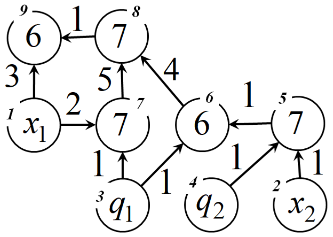

Figure 2. The network operator graph of the mathematical expression.

On the network operator graph the nodes contain the numbers of functions with two arguments, the source-nodes contain the arguments of the mathematical expression, the arcs are marked with the numbers of functions with one argument.

In the computer memory the network operator graph is presented as an integer matrix.

$$\Psi=\begin{bmatrix}0&0&0&0&0&0&2&0&3\\0&0&0&0&1&0&0&0&0\\0&0&0&0&0&1&1&0&0\\0&0&0&0&1&0&0&0&0\\0&0&0&0&7&1&0&0&0\\0&0&0&0&0&6&0&4&0\\0&0&0&0&0&0&7&5&0\\0&0&0&0&0&0&0&7&1\\0&0&0&0&0&0&0&0&6\end{bmatrix}. \tag{43}$$

Each row of the matrix corresponds to a graph node. The numbers of nodes are located at the top of the nodes (see Figure 2). The nodes are numbered in such a way that the number of the node from which the arc exits must be less than the number of the node where the arc enters. Then a network operator matrix has an upper triangular form. In the network operator matrix the numbers of functions with two arguments are located on the main diagonal. Zero element on the main diagonal shows that the row corresponds to a source-node. Other non-zero non-diagonal elements are the numbers of functions with one argument.

Consider one more symbolic regression method. In Cartesian genetic programming [25] the code of the mathematical expression (39) has the following form

$$\left(\begin{bmatrix}7\\1\\3\end{bmatrix},\begin{bmatrix}2\\5\\1\end{bmatrix},\begin{bmatrix}5\\6\\2\end{bmatrix},\begin{bmatrix}3\\1\\3\end{bmatrix},\begin{bmatrix}7\\2\\4\end{bmatrix},\begin{bmatrix}6\\9\\3\end{bmatrix},\begin{bmatrix}4\\10\\1\end{bmatrix},\begin{bmatrix}7\\11\\7\end{bmatrix},\begin{bmatrix}6\\8\\12\end{bmatrix}\right). \tag{44}$$

Here, every integer vector corresponds to one elementary function. The first element of vector is the function number, other elements are element numbers from the set of arguments (40). If a function has one argument, then the second argument is not used. The result of calculation according to each vector is added to the argument set, so a number of arguments increases in one every time after calculation of functions according to the vectors.

Due to redundancy, Cartesian genetic programming code has the same length for all mathematical expressions. A crossover operation for Cartesian genetic programming are performed by exchange of vectors after crossover point, so the length of codes does not change.

Studies of symbolic regression methods for the control synthesis problems show that it is effective to use in these methods the principle of small variations of the basic solution [26]. According to this principle only one basic solution is encoded by a method of symbolic regression. Other possible solutions are encoded by sets of variation vectors. Each variation vector makes one small change of basic solution code. After some generations the basic solution is changed on the best current found solution. This approach makes it possible to speed up the search process by narrowing the search space and avoiding additional checks for the correctness of the codes of possible solutions.

5. Computational Experiment

In a computational experiment a problem of general synthesis of optimal control for a spacecraft landing on the surface of the Moon is considered [27]. The differential equations of spacecraft state are the following

$$\begin{aligned}
\dot{x}_1 &= \frac{g_E(P_c+u_2)\cos(u_1-x_2)}{x_5} - g_{Mh}\cos(x_2),\\
\dot{x}_2 &= \frac{g_E(P_c+u_2)\sin(u_1-x_2)}{x_5} + \frac{g_{Mh}\sin(x_2)}{x_1},\\
\dot{x}_3 &= \frac{x_1\cos(x_2)}{1000},\\
\dot{x}_4 &= \frac{x_1\sin(x_2)}{1000},\\
\dot{x}_5 &= -\frac{P_c+u_2}{P_s},
\end{aligned} \tag{45}$$

where $\mathbf{x} = [x_1\ x_2\ x_3\ x_4\ x_5]^T$ is a state vector, namely x_1 is the current speed of the spacecraft (m/s), x_2 is a trajectory inclination angle (rad), x_3 is the current flight altitude relative to the lunar surface (km), x_4 is a flight distance (km), x_5 is the mass of spacecraft including fuel (kg). $\mathbf{u} = [u_1\ u_2]^T$ is a control vector, values of which are constrained

$$-\frac{\pi}{2} \le u_1 \le \frac{\pi}{2},\quad -80 \le u_2 \le 80. \tag{46}$$

Parameters of model have the following values: gravitational acceleration at the certain altitude above the lunar surface

$$g_{Mh} = g_M\left(\frac{r_M}{r_M+x_3}\right)^2, \tag{47}$$

the Moon gravitational acceleration $g_M = 1.623\ \mathrm{m/s^2}$, the Earth gravitational acceleration $g_E = 9.80665\ \mathrm{m/s^2}$, the Moon radius $r_M = 1737$ km, nominal thrust of the spacecraft engine $P_c = 720$ kg, spacecraft engine thrust $P_s = 319$ s.

A domain of initial states is

$$\mathrm{X}_0 = \{x_{0,1} = 1689,\quad -1.65 \le x_{0,2} \le -1.55,\quad 17 \le x_{0,3} \le 20,\quad x_{0,4} = 0,\quad x_{0,5} = 1500\}. \tag{48}$$

A terminal state is

$$\mathbf{x}^f = \left[x_1^f = 10,\ \ x_3^f = 0.2\right]^T. \tag{49}$$

Phase constraints are determined by the mechanics of spacecraft flight. Obviously the speed x_1, altitude x_3 and fuel level x_5 cannot be negative, reaching a zero altitude x_3 or zero fuel level x_5 at a significant speed x_1 means that the spacecraft has crashed. Consider the following phase constraints

$$\begin{array}{rcl} h_k(\mathbf{x}) & = & -x_j \le 0,\ \ k = 1,2,3,\ j = 1,3,5, \\ h_k(\mathbf{x}) & = & \vartheta(0.001 - x_j)(x_1 - V_{max}) \le 0,\ \ k = 4,5,\ j = 3,5, \end{array} \tag{50}$$

where V_{max} is the maximum landing speed, $V_{max} = 1$, $\vartheta(A)$ is the Heaviside function.

According to the proposed method at the first step the training set is to be formed. We determine the finite set of initial states within the domain (48) and solve the optimal control problem for each initial state from this set.

Let us replace the domain of initial states (48) with a set of $M = 21$ elements uniformly distributed on this domain

$$\tilde{\mathrm{X}}_0 = \left\{\mathbf{x}^{0,j+7(i-1)} = [1689\ \ -1.65 + 0.05(i-1)\ \ 17 + 0.5(j-1)\ \ 0\ \ 1500]^T\right\},\ \ i = \overline{1,3},\ \ j = \overline{1,7}. \tag{51}$$

Quality criterion considers the proximity of reaching terminal state and the case of phase constraints violation

$$\begin{array}{l} J_j = \alpha_1 \sqrt{\left(x_1\left(t_f(\mathbf{x}^{0,j})\right) - x_1^f\right)^2 + \left(x_3\left(t_f(\mathbf{x}^{0,j})\right) - x_3^f\right)^2} + \\ \alpha_2 \displaystyle\int_0^{t_f(\mathbf{x}^{0,j})} \left(\sum_{k=1}^{K} \vartheta\left(h_k(\mathbf{x}(t,\mathbf{x}^{0,j}))\right) h_k(\mathbf{x}(t,\mathbf{x}^{0,j}))\right) dt \to \min, \end{array} \tag{52}$$

where α_i, $i = 1,2$ are given penalty factors, $K = 5$ is a number of phase constraints, $j = \overline{1,M}$.

To search for solution to the optimal control problem the direct approach was used. The original problem was reduced to a nonlinear programming problem by introducing the time interval Δt. The solution of each optimal control problem in form of control vector at discrete moments of time was searched independently by hybrid evolutionary algorithm combining modern Grey Wolf Optimizer (GWO), which does not require problem specific tuning of additional parameter, and well-known Particle swarm optimization (PSO). Separately these algorithms showed a high efficiency in solving optimal control problems. A hybrid realization is to increase their effectiveness.

In a computational experiment the size of the set of possible solutions was 100, number of search iterations was 5000. Modeling parameters were the following: maximum control time $t_{max} = 300$, discretization time interval $\Delta t = 30$, penalty factors $\alpha_1 = 10$, $\alpha_2 = 10$.

At the second step of proposed approach we use obtained optimal trajectories to synthesize a multidimensional control function of object state space. The search for a control function is conducted by a symbolic regression method that search for the most suitable expression that approximates provided optimal trajectories best.

We used the network operator method to synthesize a control function. NOP allows to search for the structure of mathematical expression simultaneously with the search for optimal values of parameter vector. In the computational experiment we used the following parameters of NOP: size of NOP matrix was 40, size of the set of input variables was 3, size of the set of input parameters was 12, number of outputs was 2, number of candidate solutions in the initial set was 256, maximum number of search iteration was 25,000.

As a result of computational experiment, a control function in the form of NOP matrix and a parameter vector was obtained. The mathematical expression for the found control function is as follows

$$u_1 = \begin{cases} \pi/2, \text{ if } \tilde{u}_1 > \pi/2 \\ -\pi/2, \text{ if } \tilde{u}_1 < -\pi/2 \\ \tilde{u}_1, \text{ otherwise} \end{cases}, \quad u_2 = \begin{cases} 80, \text{ if } \tilde{u}_2 > 80 \\ -80, \text{ if } \tilde{u}_2 < -80 \\ \tilde{u}_2, \text{ otherwise} \end{cases}, \tag{53}$$

where

$$\tilde{u}_1 = \chi_6(-z_{34}, \tanh(z_{35}), z_{37}),$$

$$\begin{gathered} \tilde{u}_2 = \operatorname{sgn}(u_1)\sqrt{|u_1|} - z_{36} + \arctan(z_{35}) + \operatorname{sgn}(z_{34})\sqrt{|z_{34}|} + \log(|z_{33}|) + \\ z_{32}^{-1} + z_{31} + z_{28} - z_{28}^3 + \log(|z_{26}|) - z_{25} + \arctan(z_{21}) + \operatorname{sgn}((z_{17}))\sqrt{|z_{17}|} + \\ \exp(q_{12}) + \tanh(q_{10}) + q_9 + \log(q_5) + \tanh(q_1), \end{gathered}$$

$$z_{37} = \min\{z_{36} - z_{36}^3, \log(|z_{35}|), z_{34} - z_{34}^3, z_{32}^{-1}, \operatorname{sgn}(z_{28})\sqrt{|z_{28}|}, \tanh(z_{27}), \exp(z_{25}), z_{20}^{-1}, \sqrt[3]{x_2}\},$$

$$z_{36} = \min\{\exp(z_{29}), \tanh(z_{28}), \operatorname{sgn}(z_{27})\sqrt{|z_{27}|}, \arctan(z_{26}), \sqrt[3]{z_{22}}, z_{20}^3, \tanh(q_6), \sqrt[3]{q_1}, x_3^3\},$$

$$z_{35} = \arctan(z_{23}) + \tanh(z_{22}) + \tanh(q_{12}) + \log(|x_3|),$$

$$z_{34} = \max\{\operatorname{sgn}(z_{33})\sqrt{|z_{33}|}, \log(|z_{30}|), z_{29}^{-1}, z_{22}^{-1}, z_{20}^{-1}\},$$

$$z_{33} = \min\{z_{32}^{-1}, \arctan(z_{29}), \tanh(z_{24}), z_{22}^3, z_{19}, -z_{16}, z_{11}^{-1}, \tanh(q_3), \tanh(q_2), \exp(x_2)\},$$

$$z_{32} = \min\{z_{31}^3, z_{26}, \log(|z_{25}|), z_{18}^{-1}, \sqrt{q_{10}}, \arctan(q_6), \sqrt[3]{q_2}\},$$

$$\begin{gathered} z_{31} = \log(|z_{27}|) + z_{26}^2 + z_{24}^2 + \arctan(z_{22}) + \sqrt[3]{z_{21}} + \exp(z_{20}) + \exp(z_{17}) + \\ z_{16}^{-1} + q_{12}^3 - q_5 + \sqrt[3]{q_2} + \sqrt[3]{x_1}, \end{gathered}$$

$$z_{30} = \max\{z_{29}, -z_{26}, \tanh(z_{25}), \arctan(z_{18}), \operatorname{sgn}(z_{16})\sqrt{|z_{16}|}, q_{11}\},$$

$$z_{29} = \chi_6(z_{28}, z_{27}^{-1}, z_{26}^3, \log(|z_{24}|), -z_{22}^3, -z_{20}, \sqrt[3]{z_{17}}, q_7, q_2^{-1}, \sqrt{q_1}),$$

$$z_{28} = \max\{z_{27}, z_{23}^{-1}, -z_{20}, \log(|z_{19}|), z_{18} - z_{18}^3, -z_{17}, \log(|z_{16}|), q_7, \log(q_5)\},$$

$$z_{27} = \chi_6(z_{24}, \arctan(z_{20}), \sqrt{q_{11}}, q_8 - q_8^3, q_6^3, -q_5, \log(q_4)),$$

$$z_{26} = \chi_5(\sqrt[3]{z_{23}}, -z_{20}, z_{19} - z_{19}^3, -z_{18}, \operatorname{sgn}(z_{15})\sqrt{|z_{15}|}, q_{12}, \sqrt[3]{x_3}),$$

$$z_{25} = \max\{z_{23}^2, z_{22}, z_{21} - z_{21}^3, \arctan(z_{17}), q_{11}, q_9, \sqrt{q_5}\},$$

$$z_{24} = \max\{\tanh(z_{21}), z_{20} - z_{20}^3, z_{15}^3, q_{12} - q_{12}^3, q_{10}, q_8^3, \sqrt[3]{q_5}, q_4^3, \exp(x_2)\},$$

$$z_{23} = z_{20} z_{18} z_{16} q_9 \sqrt{q_7} \operatorname{sgn}(x_1)\sqrt{|x_1|},$$

$$z_{22} = \chi_6(z_{19}, z_{17}^{-1}, \exp(z_{16}), \sqrt[3]{q_{10}}, \arctan(q_9), q_8^{-1}, \sqrt{q_2}),$$

$$z_{21} = \max\{\tanh(z_{18}), q_8^2, q_7, -q_6, \sqrt{q_3}\},$$

$$z_{20} = \chi_6(-z_{19}, z_{17}, \arctan(z_{10}), q_6, -q_2, x_3^{-1}),$$

$$z_{19} = z_{16} + \arctan(q_{11}) + q_8^{-1} + q_5 + \arctan(x_3),$$

$$z_{18} = \chi_5(z_{15}, q_{12}^{-1}, q_7^{-1}, q_4, x_1 - x_1^3),$$

$$z_{17} = q_3 + x_3 - q_3 x_3,$$

$$z_{16} = q_2 x_2 \tanh(q_5),$$

$$z_{15} = \operatorname{sgn}(\sqrt{q_{10}} + q_5 - q_5^3 + q_3^{-1} + q_1 + x_1)\sqrt{q_{10} + (q_5 - q^3)^2 + q_3^{-2} + q_1^2 + x_1^2},$$

$$\chi_5(a_1, a_2) = a_1 + a_2 - a_1 a_2,$$

$$\chi_5(a_1, \ldots, a_s) = \chi_5(\chi_5(a_1, \chi_5(\ldots, \chi_5(a_{s-1}, a_s) \ldots), $$

$$\chi_6(a_1,\ldots,a_s) = \mathrm{sgn}\left(\sum_{i=1}^{s} a_i\right)\sqrt{\sum_{i=1}^{s} a_i^2},$$

$$q_1 = 2.3474,\ q_2 = 10.5066,\ q_3 = 9.9106,\ q_4 = 13.1419,\ q_5 = 9.6631,\ q_6 = 4.4541,$$

$$q_7 = 2.1899,\ q_8 = 4.8552,\ q_9 = 3.1116,\ q_{10} = 6.6172,\ q_{11} = 12.6812,\ q_{12} = 15.6148.$$

Functions χ_5 and χ_6 are commutative, associative, and have a unit element, zero.

To check the solution we used the found control function to obtain optimal control and corresponding trajectories for various initial states from (48). Among considered initial states were both those that were present in the training set (51) and those that were not present.

Table 1 shows the values of quality criterion J^* obtained using the found control function (53) for 21 initial states from the finite set (51). The optimal trajectories known for these initial states were previously used as a training set. This test is to show the quality of approximation. The value of the quality criterion J_{ocp} obtained by solving the optimal control problem for the same initial state is showed in the table as a reference value. The average deviation of the quality criterion values from the reference ones is 0.0591, maximum deviation is 0.2514, the standard deviation is 0.0648.

Table 1. Results of the computational experiment using initial states from the training set.

Initial State $\mathbf{x}^0$	J^*	J_{ocp}	Initial State $\mathbf{x}^0$	J^*	J_{ocp}
$[1689\ -1.65\ \ 17\ \ 0\ \ 1500]^T$	0.1777	0.0018	$[1689\ -1.55\ \ 18.5\ \ 0\ \ 1500]^T$	0.2525	0.0011
$[1689\ -1.6\ \ 17\ \ 0\ \ 1500]^T$	0.0295	0.0056	$[1689\ -1.65\ \ 19\ \ 0\ \ 1500]^T$	0.0240	0.0012
$[1689\ -1.55\ \ 17\ \ 0\ \ 1500]^T$	0.0049	0.0029	$[1689\ -1.6\ \ 19\ \ 0\ \ 1500]^T$	0.0501	0.0024
$[1689\ -1.65\ \ 17.5\ \ 0\ \ 1500]^T$	0.1433	0.0060	$[1689\ -1.55\ \ 19\ \ 0\ \ 1500]^T$	0.0030	0.0009
$[1689\ -1.6\ \ 17.5\ \ 0\ \ 1500]^T$	0.0264	0.0044	$[1689\ -1.65\ \ 19.5\ \ 0\ \ 1500]^T$	0.1035	0.0036
$[1689\ -1.55\ \ 17.5\ \ 0\ \ 1500]^T$	0.0127	0.0024	$[1689\ -1.6\ \ 19.5\ \ 0\ \ 1500]^T$	0.0822	0.0027
$[1689\ -1.65\ \ 18\ \ 0\ \ 1500]^T$	0.0780	0.0033	$[1689\ -1.55\ \ 19.5\ \ 0\ \ 1500]^T$	0.0045	0.0045
$[1689\ -1.6\ \ 18\ \ 0\ \ 1500]^T$	0.0439	0.0084	$[1689\ -1.65\ \ 20\ \ 0\ \ 1500]^T$	0.0954	0.0036
$[1689\ -1.55\ \ 18\ \ 0\ \ 1500]^T$	0.0061	0.0023	$[1689\ -1.6\ \ 20\ \ 0\ \ 1500]^T$	0.0635	0.0052
$[1689\ -1.65\ \ 18.5\ \ 0\ \ 1500]^T$	0.0703	0.0019	$[1689\ -1.55\ \ 20\ \ 0\ \ 1500]^T$	0.0334	0.0099
$[1689\ -1.6\ \ 18.5\ \ 0\ \ 1500]^T$	0.0111	0.0019			

Table 2 shows the values of quality criterion J^* obtained using the found control function (53) for 10 initial states generated randomly within the domain (48). This test is to show the suitability of the found control function for any initial state from the domain (48). The value of the quality criterion J_{ocp} obtained by solving the optimal control problem for the same initial state is showed in the table as a reference value. The average deviation of the quality criterion values from the reference ones is 0.0366, maximum deviation is 0.1122, the standard deviation is 0.0341.

Table 2. Results of the computational experiment for random initial states.

Initial State $\mathbf{x}^0$	J^*	J_{ocp}	Initial State $\mathbf{x}^0$	J^*	J_{ocp}
$[1689\ -1.565\ \ 18.92\ \ 0\ \ 1500]^T$	0.0176	0.0034	$[1689\ -1.628\ \ 18.2\ \ 0\ \ 1500]^T$	0.0398	0.0016
$[1689\ -1.571\ \ 17.21\ \ 0\ \ 1500]^T$	0.0048	0.0018	$[1689\ -1.614\ \ 19.24\ \ 0\ \ 1500]^T$	0.0508	0.0022
$[1689\ -1.63\ \ 19.39\ \ 0\ \ 1500]^T$	0.1136	0.0014	$[1689\ -1.644\ \ 18.57\ \ 0\ \ 1500]^T$	0.0613	0.0052
$[1689\ -1.558\ \ 19.91\ \ 0\ \ 1500]^T$	0.0567	0.0083	$[1689\ -1.563\ \ 18.33\ \ 0\ \ 1500]^T$	0.0032	0.0023
$[1689\ -1.582\ \ 17.62\ \ 0\ \ 1500]^T$	0.0042	0.0016	$[1689\ -1.589\ \ 18.9\ \ 0\ \ 1500]^T$	0.0464	0.0043

Figures 3 and 4 illustrate the results of computational experiment for initial state $\mathbf{x}^0 = [1689\ -1.565\ 18.92\ 0\ 1500]^T$. Graphs of the trajectories obtained using the found control function are shown by black solid lines. For comparison the trajectories obtained by solving the optimal control problem are shown by grey dashed lines. Figure 5 shows the found control function values over time.

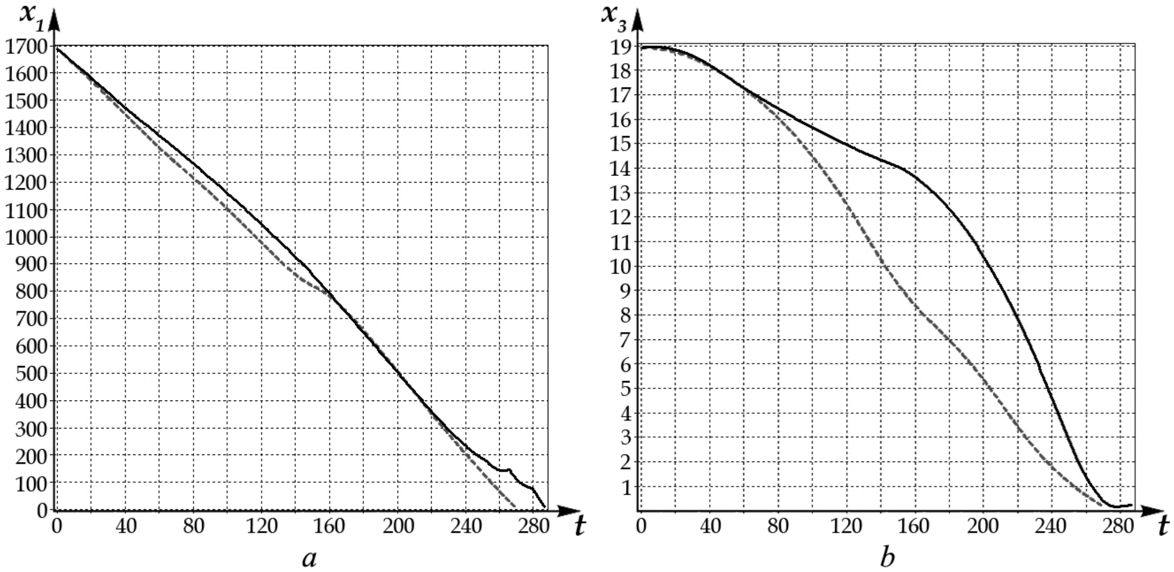

Figure 3. The graphs of trajectories: (**a**) spacecraft speed over time $x_1(t)$; (**b**) spacecraft altitude over time $x_3(t)$. Found solution—black solid line; reference solution—grey dashed line.

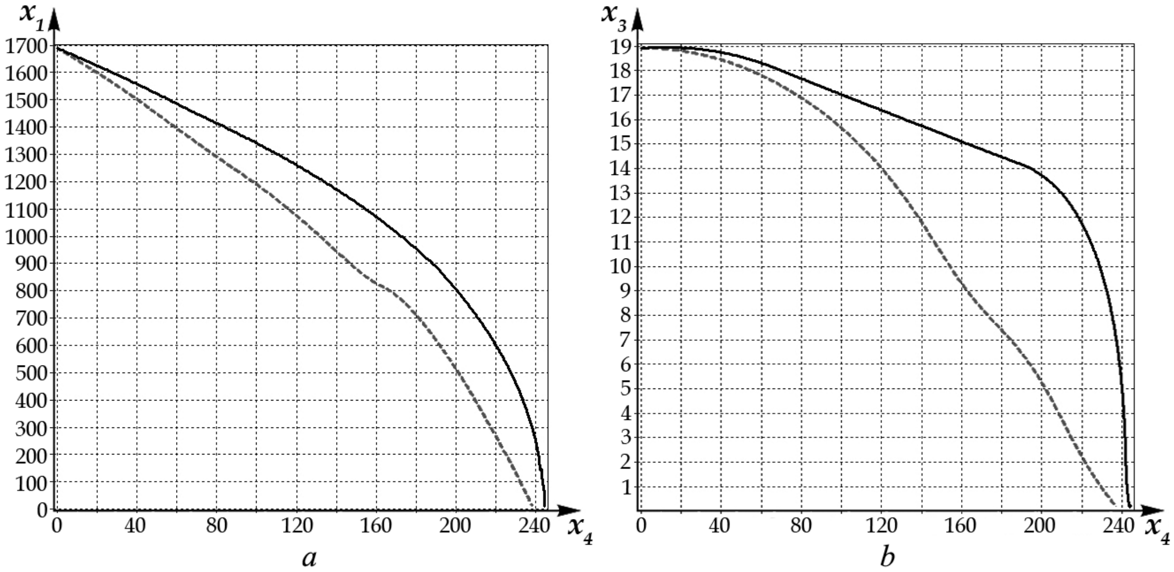

Figure 4. The graphs of trajectories: (**a**) spacecraft speed over distance $x_1(x_4)$; (**b**) spacecraft altitude over distance $x_3(x_4)$. Found solution—black solid line; reference solution—grey dashed line.

The computational experiment showed that the found multidimensional control function allows one to obtain a close to optimal solution for any initial states from the given domain (48) even for those initial states that were not in the training set (51).

According to the analysis of the standard deviation, the training set contained a sufficient number of optimal trajectories. A better value of the standard deviation for the experiment with randomly distributed in (48) initial states can be explained by the fact that the set (51) had a large number of initial states on the boundaries of the set (48).

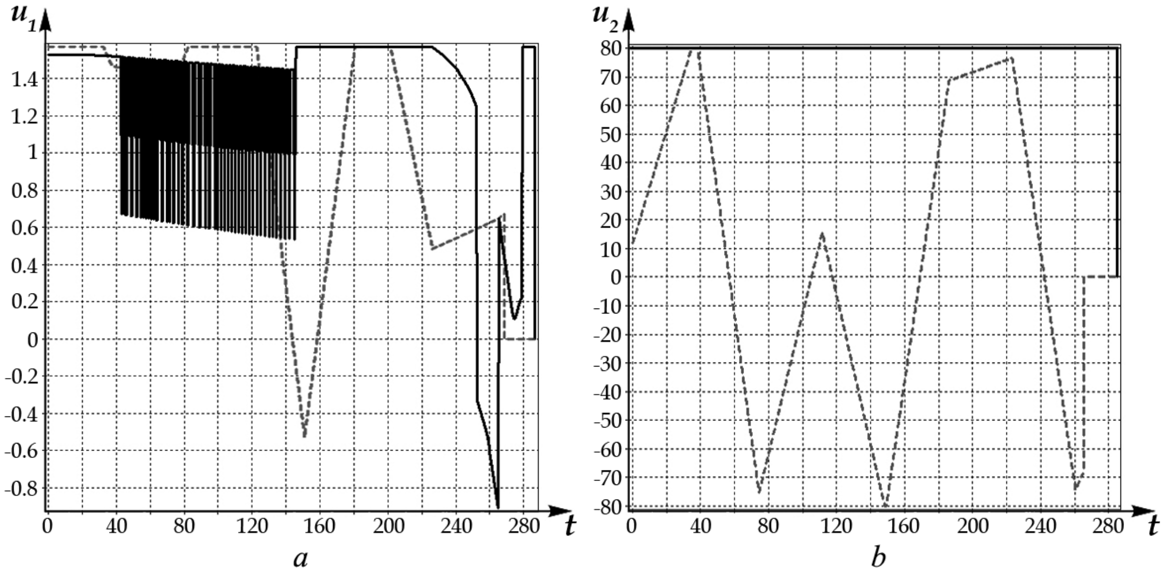

Figure 5. The graphs of control values over time: (**a**) control $u_1(t)$; (**b**) control $u_2(t)$. Found solution—black solid line; reference solution—grey dashed line.

6. Conclusions and Perspectives

The paper provides mathematical formulations of the machine learning problem, supervised and unsupervised, defines the basic concepts, such as the work area and the training set. Based on the presented formulations, it is shown that the main task of machine learning is to find a function that determines the correspondence between the input data and the resulting data. It is shown that today this problem can be solved numerically using symbolic regression methods. The problem of obtaining a mathematical expression arises in various situations—approximation of experimental data to determine a physical law or a trend model; efficiently analyze and predict variables or indicators based on previous observations; identification of a mathematical model of a process or a dynamic object; generalization of the control law based on the current state of the control object. The application of machine learning based on symbolic regression methods to control opens up the possibility of solving such a complex problem in control theory as the problem of general control synthesis. The paper presents a mathematical formulation of the control synthesis problem and provides methods for its solution using machine learning both directly and based on a training set. An important result of the article is the methodology for solving the problem of general control synthesis as machine learning control based on a training set. An approach to constructing a training sample based on multiple solutions to the optimal control problem is proposed. An example of solving a specific problem of synthesis of control of a complex technical object based on the approximation of optimal trajectories is given. It is shown that such a control, obtained on the basis of machine learning, gives good results not only for the input data from the training set, but also not from it.

The concept of machine learning is widely known, but very often limited by its association with neural network technology. We are expanding the concept of machine learning to include a description of an unknown function in its formulation. Thus, a function can be specified and training is aimed only at finding parameters, as in neural networks, but you can also search for the structure of the function and its parameters. This became possible with the advent of symbolic regression methods. The complexity of these methods lies in the need to organize search in a space in which there is no metric. This greatly complicates the solution of the problem of finding the required structure of the function. This complexity opens up a wide field of research. One of the ways to solve this problem is to use the principle of small variations of the basic solution indicated in the article. This approach allows you to concentrate the search for a solution around a basic solution based on the developer's experience or intuition. This approach also requires further study.

Author Contributions: Conceptualization, A.D. and E.S.; methodology, A.D., E.S. and S.K.; software, S.K.; validation, E.S. and G.D.; formal analysis, A.D., G.D.; investigation, S.K.; data curation, S.K.; writing—original draft preparation, A.D., E.S. and S.K.; writing—review and editing, E.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the Ministry of Science and Higher Education of the Russian Federation, project No. 075-15-2020-799.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Duriez, T.; Brunton, S.L.; Noack, B.R. *Machine Learning Control–Taming Nonlinear Dynamics and Turbulence*; Fluid Mechanics and Its Applications; Springer International Publishing Switzerland: Berlin/Heidelberg, Germany, 2017; Volume 116, 211p.
2. Alibekov, E.; Kubalık, J.; Babushka, R. Symbolic method for deriving policy in reinforcement learning. In Proceedings of the 55th IEEE Conference on Decision and Control (CDC), Las Vegas, NV, USA, 12–14 December 2016; pp. 2789–2795.
3. Levine, S.; Koltun, V. Learning complex neural network policies with trajectory optimization. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 829–837.
4. Nagabandi, A.; Kahn, G.; Fearing, R.S.; Levine, S. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 7559–7566
5. Gao, T.; Liu, Y.; Liu, L.; Li, D. Adaptive neural network-based control for a class of nonlinear pure-feedback systems with time-varying full state constraints. *IEEE/CAA J. Autom. Sin.* **2018**, *5*, 923–933. [CrossRef]
6. Koza, J.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*; MIT Press: Cambridge, MA, USA; London, UK, 1992; 819p.
7. Dracopoulos, D.; Kent, S. Genetic Programming for Prediction and Control. *Neural Comput. Appl.* **1998**, *6*, 214–228. [CrossRef]
8. Dracopoulos, D. Genetic Algorithms and Genetic Programming for Control. In *Evolutionary Algorithms in Engineering Applications*; Springer: Berlin/Heidelberg, Germany, 1997; pp. 329–343.
9. Boltyanskii, V.G. *Mathematical Methods of Optimal Control*; Holt, Rinehart and Winston: New York, NY, USA, 1971; 272p.
10. Krstic, M.; Kanellakopoulos, I.; Kokotovic, P.V. *Nonlinear and Adaptive Control Design*; Wiley–Interscience: New York, NY, USA, 1995; 576p.
11. Khalil, H.K. *Nonlinear Control*; Pearson Education, Inc.: Prentice Hall, NJ, USA, 2015; 403p.
12. Kolesnikov, A.A. Introduction of synergetic control. In Proceedings of the American Control Conference ACC-2014, Portland, OR, USA, 4–6 June 2014; pp. 3013–3016.
13. Kolesnikov, A.; Veselov, G.; Kolesnikov, A.; Monti, A.; Ponci, F.; Santi, E.; Dougal, R. Synergetic synthesis of DC-DC boost converter controllers: Theory and experimental analysis. In Proceedings of the IEEE Applied Power Electronics Conference and Exposition—APEC, Dallas, TX, USA, 10–14 March 2002; pp. 409–415.
14. Diveev, A.I. A Numerical Method for Network Operator for Synthesis of a Control System with Uncertain Initial Values. *J. Comput. Syst. Sci. Int.* **2012**, *51*, 228–243. [CrossRef]
15. Diveev, A.I.; Ibadulla, S.I.; Konyrbaev, N.B.; Shmalko, E.Y. Variational Genetic Programming for Optimal Control System Synthesis of Mobile Robots. *IFAC-PapersOnLine* **2015**, *48*, 106–111. [CrossRef]
16. Diveev, A.I.; Ibadulla, S.I.; Konyrbaev, N.B.; Shmalko, E.Y. Variational Analytic Programming for Synthesis of Optimal Control for Flying Robot. *IFAC-PapersOnLine* **2015**, *48*, 75–80. [CrossRef]
17. Diveev, A.I.; Shmalko, E.Y. Optimal Motion Control for Multi-Robot System by Multilayer Network Operator. In Proceedings of the 11th IEEE Conference on Industrial Electronics and Applications (ICIEA 2016), Hefei, China, 5–7 June 2016; pp. 2168–2173.
18. Diveev, A.I.; Balandina, G.I.; Konstantinov, S.V. Binary Variational Genetic Programming for the Problem of Synthesis of Control System. In Proceedings of the 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD 2017), Guilin, China, 29–31 July 2017; pp. 165–170.
19. Diveev, A.I. Cartesian Genetic Programming for Synthesis of Optimal Control System. In Proceedings of the Future Technologies Conference (FTC) 2020, Vancouver, BC, Canada, 5–6 November 2020; Springer Nature Switzerland AG: Cham, Switzerland, 2021; Volume 2, pp. 205–222.
20. Diveev, A.I.; Shmalko, E.Y.; Sofronova, E.A. Theoretical Fundamentals for Unimodality Estimation of an Objective Functional in the Optimal Control Problem. In Proceedings of the 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), Paris, France, 23–26 April 2019; pp. 767–772.
21. Diveev, A.I.; Konstantinov, S.V. Study of the Practical Convergence of Evolutionary Algorithms for the Optimal Program Control of a Wheeled Robot. *J. Comput. Syst. Sci. Int.* **2018**, *57*, 561–580. [CrossRef]
22. Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison–Wesley: Reading, MA, USA, 1989.

23. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
24. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
25. Miller, J.; Thomson, P. Cartesian Genetic Programming. In *Proceedings EuroGP 2000R 3rd European Conf. Genetic Programming*; Poli, R., Banzhaf, W., Langdon, W.B., Miller, J.F., Nordin, P., Fogarty, T.C., Eds.; Springer: Edinburgh, UK; Berlin, Germany, 2000; Volume 1802, pp. 121–132.
26. Diveev, A.I. Small Variations of Basic Solution Method for Nonnumerical Optimization. *IFAC-PapersOnLine* **2015**, *48*, 28–33. [CrossRef]
27. Liu, X.L.; Duan, G.R.; Teo, K.L. Optimal Soft Landing Control for Moon Lander. *Automatica* **2008**, *44*, 1097–1103. [CrossRef]